Testing Euclidean Spanners

F. Hellweg, M. Schmidt, C. Sohler

06.09.2010

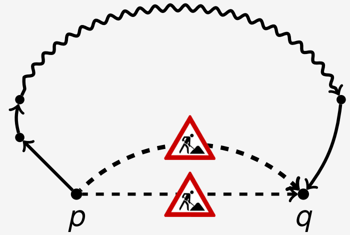
Road works!

Road works!

Road works!

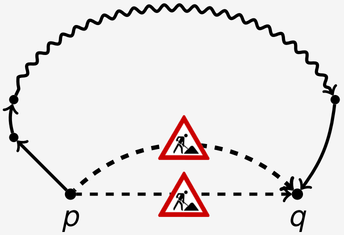
Road works!

Road works!



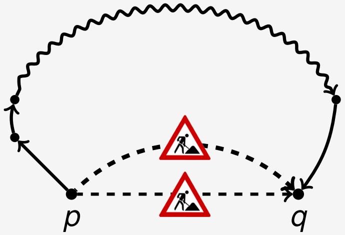
Road works!

Let $0 < \delta < 1$ and P a set of points in $\mathbb{R}^d$ for constant d.



Road works!

Let $0 < \delta < 1$ and P a set of points in $\mathbb{R}^d$ for constant d.



Is there a violating pair of points $p, q \in P$ such that

$$d_G(p, q) > (1 + \delta)||q - p||_2 \text{ ?}$$

Road works!

Let $0 < \delta < 1$ and P a set of points in $\mathbb{R}^d$ for constant d.

Definition (Euclidian $(1+\delta)$-spanner)

A directed geometric graph $G = (P, E)$ is a $(1+\delta)$-*spanner*, if

$$\forall p, q \in P : d_G(p, q) \leq (1+\delta)||q - p||_2$$

where $d_G(p, q)$ denotes the graph distance between p and q.

Is there a violating pair of points $p, q \in P$ such that

$$d_G(p, q) > (1+\delta)||q - p||_2 \text{ ?}$$

Euclidean Spanners and Property Testing

Wireless ad-hoc networks

- changing network structure
- do the established links still satisfy the spanner property?
- fast test required

Wireless ad-hoc networks

- changing network structure
- do the established links still satisfy the spanner property?
- fast test required

Aim

Find a sublinear testing algorithm that

- always accepts $(1 + \delta)$-spanners
- rejects graphs that are very far away from being a spanner
- (deals with other graphs as it likes)
- returns a violating pair when rejecting a graph

Wireless ad-hoc networks

- changing network structure
- do the established links still satisfy the spanner property?
- fast test required

Aim

Find a sublinear testing algorithm that

- always accepts $(1+\delta)$-spanners
- rejects graphs that are very far away from being a spanner
- (deals with other graphs as it likes)
- returns a violating pair when rejecting a graph

Sublinear $\rightsquigarrow$ randomized algorithm with (small) error probability.

Geometric Property Testing

Geometric Property Testing

Property Testing

- provides a relaxation of decision problems
- introduced by Rubinfeld and Sudan (1996)
- study of combinatorial properties:
  Goldreich, Goldwasser, and Ron (1998)

Geometric Property Testing

Property Testing

- provides a relaxation of decision problems
- introduced by Rubinfeld and Sudan (1996)
- study of combinatorial properties:
  Goldreich, Goldwasser, and Ron (1998)

Geometric Properties

- Euclidean minimum spanning tree:
  Czumaj, Sohler, Ziegler (2000)
- Clusterability of point sets: Alon, Dar, Parnas, Ron (2003)
- Convexity: Rademacher, Vempala (2004)

ε-farness

ε-farness

Let $0 < \varepsilon < 1$ and $0 < \delta < 1$.

Definition

A directed geometric graph $G = (P, E)$ is ε-far from a directed geometric graph $G' = (P, E')$ if

$$|E \backslash E' \cup E' \backslash E| > \varepsilon n.$$

G is ε-far from having the property to be a $(1+\delta)$-spanner if it is ε-far from every $(1+\delta)$-spanner.

ε-farness

Let $0 < \varepsilon < 1$ and $0 < \delta < 1$.

Definition

A directed geometric graph $G = (P, E)$ is ε-far from a directed geometric graph $G' = (P, E')$ if

$$|E \backslash E' \cup E' \backslash E| > \varepsilon n.$$

G is ε-far from having the property to be a $(1 + \delta)$-spanner if it is ε-far from every $(1 + \delta)$-spanner.

ε-farness

Let $0 < \varepsilon < 1$ and $0 < \delta < 1$.

Definition

A directed geometric graph $G = (P, E)$ is ε-far from a directed geometric graph $G' = (P, E')$ if

$$|E \backslash E' \cup E' \backslash E| > \varepsilon n.$$

G is ε-far from having the property to be a $(1 + \delta)$-spanner if it is ε-far from every $(1 + \delta)$-spanner.

Aim

Find a sublinear testing algorithm that

- always accepts $(1+\delta)$-spanners
- rejects graphs ε-far from being a spanner w.h.p.
- (deals with other graphs as it likes)
- returns a violating pair when rejecting a graph

Challenge: Find one of εn violating pairs with few sample points

Aim

Find a sublinear testing algorithm that

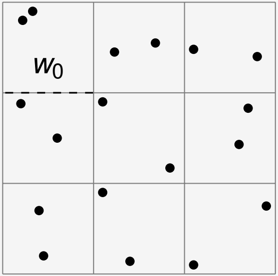
- always accepts $(1+\delta)$-spanners
- rejects graphs ε-far from being a spanner w.h.p.
- (deals with other graphs as it likes)
- returns a violating pair when rejecting a graph

Challenge: Find one of εn violating pairs with few sample points

Note

- not necessarily εn distinct points
- compared to $\Omega(n^2)$ possible pairs of points, εn is small

Special case



G: directed geometric graph

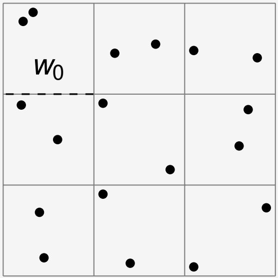
n_u: value depending on n, $1/\delta$ and $1/\varepsilon$

H: d-dimensional grid

- containing $\mathcal{O}(n_u)$ points of G in each cell
- covering G completely

We say G is uniformly distributed with parameter n_u.

Special case



G: directed geometric graph

n_u: value depending on n, $1/\delta$ and $1/\varepsilon$

H: d-dimensional grid

- containing $\mathcal{O}(n_u)$ points of G in each cell
- covering G completely

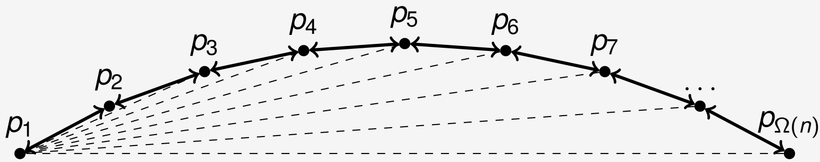
We say G is uniformly distributed with parameter n_u.

$\rightsquigarrow$ for every point p, number of points close to p is $\approx n_u$

Coping with far-away errors

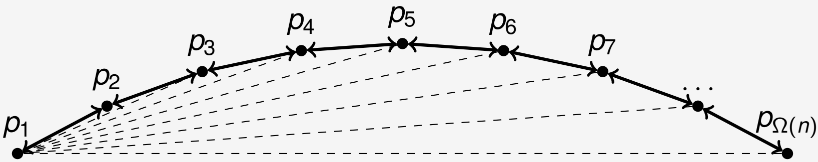
Coping with far-away errors

Distant violating pairs



Coping with far-away errors

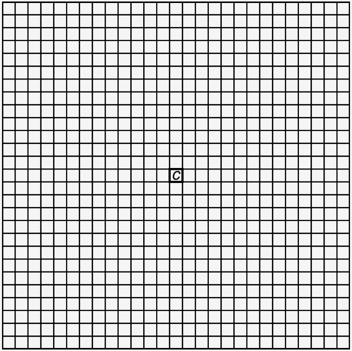
Distant violating pairs



Find a violating pair

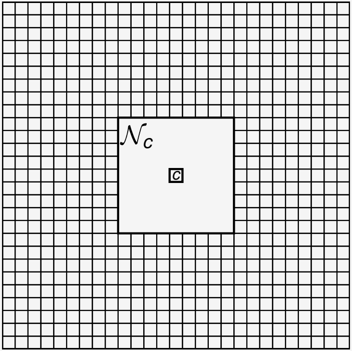
- ε-far $\rightsquigarrow$ at least εn violating pairs
- establish property for all distant pairs with $\varepsilon n/2$ edges
- $\varepsilon n/2$ violating pairs with small distance remain

Coping with far-away errors



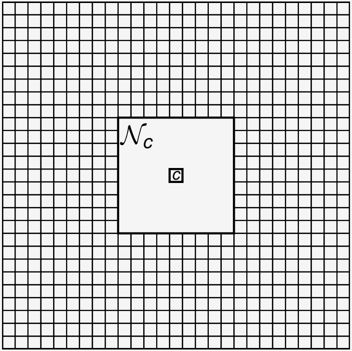
For all cells of H:

Coping with far-away errors



For all cells of H:

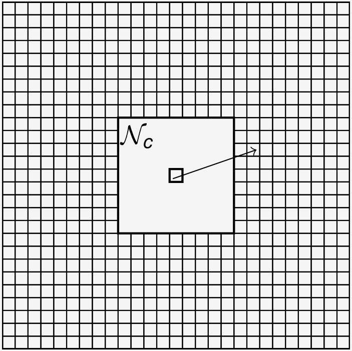
Coping with far-away errors



For all cells of H:

- define neighborhood $\mathcal{N}_c$ large enough such that distance dominates cell diameter (width $c \cdot w_0$)

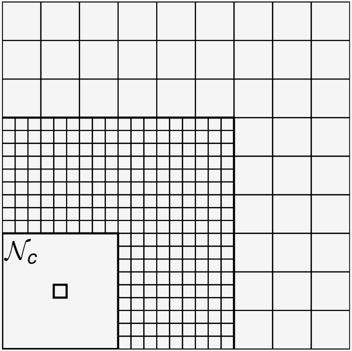
Coping with far-away errors



For all cells of H:

- define neighborhood $\mathcal{N}_c$ large enough such that distance dominates cell diameter (width $c \cdot w_0$)
- add one edge to every cell within a certain distance

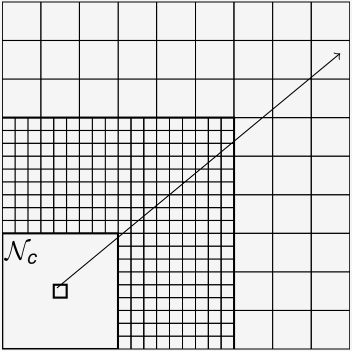
Coping with far-away errors



For all cells of H:

- define neighborhood $\mathcal{N}_c$ large enough such that distance dominates cell diameter (width $c \cdot w_0$)
- add one edge to every cell within a certain distance

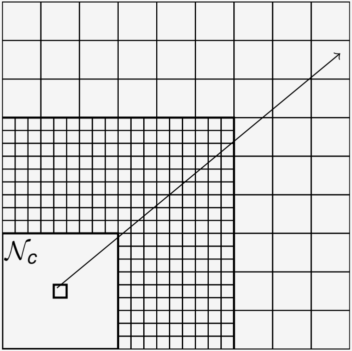
Coping with far-away errors



For all cells of H:

- define neighborhood $\mathcal{N}_c$ large enough such that distance dominates cell diameter (width $c \cdot w_0$)
- add one edge to every cell within a certain distance
- grow distance and cell diameter simultaneously

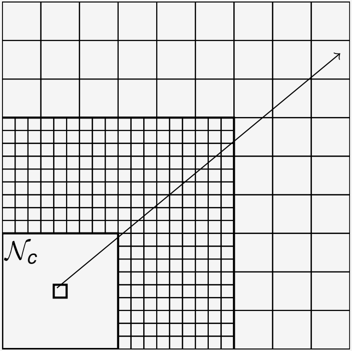
Coping with far-away errors



For all cells of H:

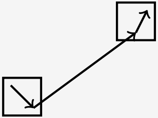
- define neighborhood $\mathcal{N}_c$ large enough such that distance dominates cell diameter (width $c \cdot w_0$)
- add one edge to every cell within a certain distance
- grow distance and cell diameter simultaneously

Coping with far-away errors



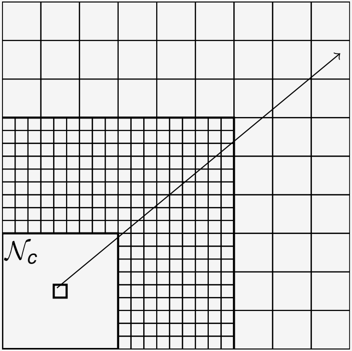
Assure the following:

- If all neighbors satisfy spanner property, extended graph is a spanner



(To prove this, recursively apply grid property)

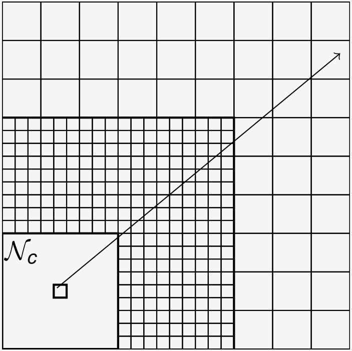
Coping with far-away errors



Assure the following:

- If all neighbors satisfy spanner property, extended graph is a spanner
- Number of cells in each level is constant
- Number of all edges in all exponential grids is at most $\varepsilon n/2$

Coping with far-away errors



Assure the following:

- If all neighbors satisfy spanner property, extended graph is a spanner
- Number of cells in each level is constant
- Number of all edges in all exponential grids is at most $\varepsilon n/2$

Exponential grids:
Har-Peled, Mazumdar (2004)

The algorithm

The algorithm

- Sample $q_s = \tilde{\mathcal{O}}(\delta^{-d}\varepsilon^{-2}\sqrt{n})$ points
- For every sample point p
  - Sample all q_n points within radius $(1+\delta)\sqrt{d} \cdot c \cdot w_0$ of p

  ⇝ if a neighbor q of p is not found, then (p, q) is violating
- Accept graph iff no violating pair was found

The algorithm

- Sample $q_s = \tilde{\mathcal{O}}(\delta^{-d}\varepsilon^{-2}\sqrt{n})$ points
- For every sample point p
  - Sample all q_n points within radius $(1+\delta)\sqrt{d} \cdot c \cdot w_0$ of p

  $\rightsquigarrow$ if a neighbor q of p is not found, then (p, q) is violating
- Accept graph iff no violating pair was found

Observation

- $n_u = \mathcal{O}(\delta^{-d}\varepsilon^{-1}\log\delta n)$

$\rightsquigarrow$ $q_n = \mathcal{O}(\delta^{-d}\varepsilon^{-1}\log\delta n)$

$\rightsquigarrow$ $q_s + q_s \cdot q_n = \tilde{\mathcal{O}}\left(\delta^{-2d}\varepsilon^{-3}D\sqrt{n}\right)$ queries

- The algorithm accepts every spanner with probability 1.
- In case of rejection, a violating pair was found.

Uniformly spread points

Problem

How do we figure out that there is a neighbor q of p that was not found during the exploration of $p's$ neighborhood?

$\rightsquigarrow$ only possible if p and q were sampled.

Problem

How do we figure out that there is a neighbor q of p that was not found during the exploration of $p's$ neighborhood?

$\rightsquigarrow$ only possible if p **and** q were sampled.

Birthday paradox type argument

- to sample $\sqrt{n}$ disjoint points p having a neighbor q such that (p, q) is violating, only $\tilde{\mathcal{O}}(\varepsilon^{-1}\sqrt{n})$ samples are necessary
- this ensures that there are $\sqrt{n}$ points q such that p is sampled and (p, q) form a violating pair
- to find one of these, $\tilde{\mathcal{O}}(\sqrt{n})$ additional samples needed

Let $0 < \delta < 1$ and $0 < \varepsilon < 1$.

Theorem

Under the condition that the directed geometric graph $G = (P, E)$ is uniformly distributed with

$$n_u = \mathcal{O}(\delta^{-d}\varepsilon^{-1} \log \delta n)$$

we can decide w.h.p. whether G is an Euclidean spanner or ε-far from this property with runtime and query complexity

$$\tilde{\mathcal{O}}\left(\delta^{-2d}\varepsilon^{-3} D\sqrt{n}\right).$$

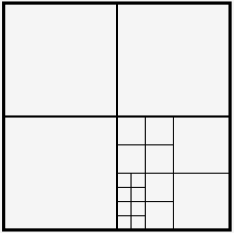
Arbitrary point sets

Generalization

- Assume point coordinates are integer and on a d-dimensional grid $\{1, \ldots, \Delta\}^d$.
- Replace grid H by a quad-tree

Problem

The neighborhood of a base cell (of the quad-tree) can now contain many cells. Such base cells are denoted as heavy cells.



- stop partitioning at logarithmic number of points

$\rightsquigarrow$ many cells can lie in the neighborhood of a large cell

Coping with heavy cells

Problem

Neighborhood of a large cell can contain many smaller cells.

Coping with heavy cells

Problem

Neighborhood of a large cell can contain many smaller cells.

Observation

A small cell can not be in the neighborhood of many larger cells. (the number is bounded from above by $\mathcal{O}(\log \Delta)$).

- $\leadsto$ bound on the number of relations of the kind 'small cell in neighborhood of larger cell'
- $\leadsto$ bound on the number of heavy cells

Lemma

If all pairs of a point p in cells that are not heavy and a neighbor q of p satisfy the spanner property, then the property can be satisfied for all remaining pairs by adding $\varepsilon n/2$ edges.

Let $0 < \delta < 1$ and $0 < \varepsilon < 1$.

Theorem

We can decide w.h.p. whether a directed geometric graph $G = (P, E)$ placed on a $\{1, \ldots, \Delta\}^d$-grid is an Euclidean spanner or ε-far from this property with runtime and query complexity

$$\tilde{\mathcal{O}}(\delta^{-5d} \epsilon^{-5} D \log^6 \Delta \sqrt{n}).$$

Open problems

- Match upper and lower bound $\left[\text{currently } \Omega\left(n^{1/3}\right)\right]$.
- Can the exponential influence of the dimension be avoided?
- What about the grid width Δ?
- Can point coordinates contribute more information?
- Which query complexity is needed to test the spanner property for different metrics?

Thank you for your attention!